import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

# Load the dataset

data = pd.read\_csv(r'C:\Users\dheer\OneDrive\Desktop\Vardhaman collage\CDC\train.csv')

# Prepare the data (assuming the first column contains labels and the rest are pixel values)

X = data.iloc[:, 1:].values # Features (pixels)

y = data.iloc[:, 0].values.astype(int) # Labels (digits), ensure labels are integers

# Normalize the pixel values (0-255 -> 0-1)

X = X / 255.0

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the Random Forest Classifier

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the model

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Calculate accuracy

test\_acc = accuracy\_score(y\_test, y\_pred)

print(f'Test accuracy: {test\_acc:.2f}')

# Plot a confusion matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(10,7))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=range(10), yticklabels=range(10))

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix')

plt.show()